SINC - LINK

Vol. 4 No. 3 MAY/JUN 1986

SINC LINC IS A PUBLICATION OF THE TORONTO TIMEX-SINCLAIR USERS CLUB AND IS ISSUED 6 TIMES A YEAR. COPIES OF THE NEWSLETTER ARE $1.50 EACH FOR NON-MEMBERS. CLUB MEMBERS RECEIVE A FREE COPY AS PART OF THE $20.00 ANNUAL MEMBERSHIP FEE.

NEWSLETTERS ARE EXCHANGED, FREE OF CHARGE, WITH OTHER TIMEX-SINCLAIR USERS GROUPS

SEND CORRESPONDANCE TO:

Attention: SINC-LINK Editor
TORONTO TIMEX-SINCLAIR USERS CLUB
P.O. Box 7274 Stn. A, Toronto M5W 1X9

NEWSLETTER INDEX

EXECUTIVE OFFICERS

PRESIDENT:	George Chambers (416) 751-7559) 14 Richome Court Scarborough, Ont M1K 2Y1
TREASURER:	Charles Urban
TAPE LIBRARIANS:	Robert Rosenzweig, Brian Milne
PAPER LIBRARIAN:	John Burns
HOUSE CHAIRMAN:	Help Wanted!!
ACTIVITY DIRECTOR:	Help Wanted!!
LIAISON OFFICER:	(Out-of-Town Members) George Chambers
NEWS EDITOR:	John Roach

PRESIDENT'S MESSAGE

This September will mark the near completion of our 4th year of club operation. It was in November 1982 that our club had it's beginnings. A lot has happened in that time. The club has grown from an original dozen members to a current membership of about 130. Where the club was originally composed of members from the Toronto Area, we now have about 55 members living outside of Toronto. In that time also we have built up an extensive paper and tape library in the club. Our newsletter has grown from it's initial 6 pages to the present 14 or 16 pages. We have moved several times in that period.

What does the future hold for us? For one thing, there is a move in the works. Present indications are that the new premises of the North York Memorial Centre will be ready in January. Coupled with this is an indication that the rent will rise by $20. This may mean an increase in dues, or else a change in premises.

One other troubling aspect is the difficulty in getting members to take an active role in the club operations. This problem is of course common to all organizations. However in the case of a self-serving organization such as ours, if there is a lack of interest in participating in the club operation it brings into question the merit of the club's existance.

Keep this in mind when the September annual meeting with it's election of officers rolls around. Offer your services.

PASSWORD ROUTINE

By using the commands "ON ERR" and "GOTO" it is possible to set up a password routine that is, if not unbeatable, at least very tough. In the listing below the "ON ERR" command, which treats both "STOP" and "BREAK" as errors, sends the computer back to line 30 if there are any errors. The program cannot be broken into or stopped, unlike if only the "INPUT" command was used. For added protection a "FOR/NEXT" loop and "NEW" command has been added. This allows for only three attempts at the password before erasing the program. The password can be any string.

```
 10 ON ERR  GO TO 30
 20 FOR x=1 TO 3
 30 INPUT "Password please";a$
 40 IF a$="QTS" THEN  GO TO 100
 50 PRINT "Incorrect, try again"
 60 PAUSE 60: CLS
 70 NEXT x
 80 NEW
 90 LIST
100 PRINT "You are now past the Password"
```

CONTROL CODES
BY Barry Lindstrom

The 2068 has a series of control codes that can be easily used by the BASIC programmer. They are listed in the owner's manual on pages 239 and 240, and deal with the screen display.

They can be used in two ways. First and easiest, is to use them in the CHR$ statement; to change the paper color to RED, for example, you would print CHR$17 + CHR$ 2. Notice that 17 is the character code for "PAPER CONTROL" and 2 is the number of RED (on your keyboard). You could even assign these to a string variable for convenience: Let A$ = CHR$ 17 + CHR$ 2, followed by a PRINT A$ would do the same thing.

The second way to use them is in a DEF FN statement. If I had names or some other information stored in string variables and sometimes wanted them to "FLASH" when I printed them, I would have DEF FN D$(N$) = CHR$ 18 + CHR$ 1 + N$ + CHR$ 18 + CHR$ 0 somewhere early in my program. The CHR$ 18 deals with "FLASH"; CHR$ 1 turns it on, and CHR$ 0 turns it back off again. To use it, print FN D$("HELLO"), or what ever the message is.

If you are careful you can combine quite a few of these control codes in a string variable (or function). Just remember to turn off any functions you turned on.

The following demonstration program shows each of the control codes that can be used. Unfortunately, the "CURSOR UP" and "CURSOR DOWN" codes do not work for me. If you figure out how to use them like the others, please let me know!!

This article was reprinted from the TAS BAM Users Group newsletter KEYBOARDS VOL.4 NO.1; JAN/FEB 1986

ZX81 TIPS

When using the first program line for entering machine code, PEEKing the address 16511 will show the number of characters in the REM line. If there are more than 256 characters in the line (a rare occurance) then enter PRINT PEEK 16511 + 256 * PEEK 16512.

Use the following line in your program to prevent an accidental screen memory overflow error.

IF PEEK 16442 < 4 then CLS (or SCROLL)

SincBits

Ian Robertson
Compuserve 72167,3401

UPDATES: Has anyone tried to return a 2068 (board only) to Timex, Little Rock, for repair? Well I have, only to have it returned, with a note saying that they DO NOT REPAIR 2068's any more!! Several of our club members have returned complete units to Timex, along with $30.00 (US), and have received refurbished units in return. The bottom line must be that they do NOT fix individual boards, but that they DO replace one complete computer with another! I would appreciate feedback on this matter. "Complete Units" means that you must send back everything that came with the computer, when you purchased it.
If you have been putting off buying a colour composite video monitor, you should look at the NEC model JC-1225-MA, presently being sold by "Computers For Less" for $199.00 (equivalent to $149.00 US). Just remember, that if you use your computer primarily for text, that a colour composite video is NOT for you. The text readability is not very good. But, on the other hand, if you use your computer for games, art work, etc., then it is GREAT. Now for a couple of interesting items gleaned from the July issue of Sinclair User. It seems that when Amstrad bought the Sinclair Computer rights, that the purchase did not include Portugal and the Eastern Bloc countries. Timex Portugal has the rights to them and has just closed a deal with the Polish government for the supply of 800,000 TC2068's and 300,000 FDD 3000 twin 3" disc drives (these are the 3" drives sold by Zebra Systems). The other item concerns production of the first "Amstrad Spectrum", to be called the Spectrum Plus Two, tentatively scheduled for release this Autumn (whenever that is). It will have a built-in cassette recorder and quite probably - a proper keyboard. Sir Clive has already established a seperate company, called Anamartic, to continue working on his wafer-scale chip project.

SPECTRUM: In my last column I was very enthusiastic about a recently aquired Spectrum utility called "Supadrive". Well, since then I have been lucky enough to purchase a similar utility called "M/DRIVE DOCTOR 3.0" from Pipeq Systems, 151 Millbridge Rd., Dollis Valley Way, Barnet, Herts, U.K. It has 13 commands which enhance your m/d computing pleasure. This is a 9.5 out of 10 product. Another such (highly rated) product is "MULTIFACE 1", from Romantic Robot, which not only saves "any" program at the push of a button, but also allows you to goto basic and add pokes, revise, etc. I have been lucky enough to be able to borrow one to experiment with and can report that not only does it do all the things claimed, but that it is also compatible with both the BETA and the KEMPSTON DDI's. If you do a lot of word processing you should consider a program by Seven Stars Publishing, 34 Squirrel Rise, Marlow, Bucks SL7 3PN, U.K. The program is called TASPRO. When merged with Tasword II it allows your 80 column printer to print out text with identical spacing between words on each line. It sells for £8.00 (pounds sterling), which includes airmail. There is also a version for Tasword III. They also have just released a program called "Qualitas", which is supposed to allow your 80 column printer to print out in NLQ. More on that in my next column. You must tell them which printer you will be using, for any program, when you order. Cameron Hayne, our member in Oxford (UK), has just finished his Basic Compiler, which he calls "TIMACHINE", and I can report from first hand experience - "THAT IT BLOWS AWAY ALL COMPETITION FROM ANY OTHER SPECTRUM COMPILER". When Cameron gets it marketed (he is currently negociating with 2 UK software houses), it will be considered A MUST BUY item. Cameron has a 2068 version presently being marketed by our own neighbourhood software house - Novelsoft, 106 Seventh Street, Toronto, Ontario M8V 3B4. The price is $20.00 (US) or $25.00 (CDN).

2068: If you have been thinking about ordering a Kempston compatible joystick from the U.K., make sure you order one which has a 46 pin connector (same as the ZX81/TS1000) and NOT the 56 pin connector common to the Spectrum. Chances are the 56 pin model will not work with the 2068. Cheetah and Datel both have such models for sale. The other way around this is to order one of the "Kempston Compatible" IF's from either JOHN OLIGER (you must have his expansion board), Zebra Systems or Russell Electronics. Latest on the John Oliger Disk Drive Interface (DDI). Version 1.32 of his DOS, called SAFE DOS, includes screen copy to printer (80 column or 2040) and allows the user to exit to basic by pressing the NMI button and then pressing "C". John says that both he and Ray Kingsley are presently working on a "full blown" business type DOS for his DDI. I have been busy putting my JLO expansion board system in a metal case, in order to reduce RFI and to improve it's appearance. The case contains the 2 boards for the DDI, printer IF and Kempston joystick IF. It also has LEDs and switches on the front, with joystick and disc connectors mounted on the back. One of the switches on the front is to select one of two Eproms mounted on Dan Roman's eprom/emulator board. The case is painted to match the 2068, and connects to the 2068 via a 3" 64 conductor ribbon cable, which has gold connectors at each end. All connections on this system are gold, for reliability. Remember that the JLO DDI is compatible with the 2068 while it is in the Spectrum mode. You should see some of those Spectrum games and Artworx, version 1.1, in colour! Remember, I am the guy that used to say "COLOUR IS NOT REALLY NECESSARY FOR THE 2068".

TS1000/ZX81: Just a reminder that some very remarkable software/hardware items are available from Fred Machbaur, Silicon Mountain Computers, C-12, Mtn. Stn. Group Box, Nelson, BC V1L 5P1, Canada. Write for a catalogue, but include a large SASE. When my JLO video upgrade was working (before the much-dreaded-static-zap) I was running Fred's JD-BASIC and can say from first hand experience that this is indeed an experience. Fred also sells fully assembled/tested video upgrades for those not so dedicated to the soldering iron. Another great new item is the LARKEN DDI, from Larken Electronics, R.R.#2, Navan, Ontario K4B 1H9, Canada. It comes as either a single drive or a double drive controller. It sells for $99.00 plus $4.00 for the drive cable and $5.00 shipping (US funds). There are either 2 or 3 club members running this system and the two that I know of are VERY HAPPY about their purchase. When ordering specify that you require either the single drive or the double drive DDI.

I must confess to being one of the culprits behind the lateness of this issue. The demands of my "real job" and other obligations have left me with virtually zero time for computer activities over the past several weeks.

Things look a little better for the rest of the year, though (I have some holiday time coming), and we should be back on schedule by the Sept-Oct issue.

2068 USERS TAKE NOTE: Will wonders never cease! Every single topic in this month's ZX81 Resources applies to your computer, too!

LARKEN DISK SYSTEM

After using the Larken for a couple of months now, I must say it meets or exceeds all expectations. The DOS (ZX-LDOS) is certainly the most sophisticated available for the ZX81, but its operation is simple, logical, and well documented. LDOS lends itself extremely well to user-written DOS utilities for damaged-track recovery, etc. The ability to Save & Load BASIC, ARRAY, or CODE files up to 47K long makes it a winner, providing programmed data manipulation capabilities never before possible. The DOS commands & disk format are IDENTICAL to the 2068 version, so CODE written on a 2068 and saved on a 2068 Larken disk may be loaded by a ZX81 from the same disk!

The system has been 100% reliable without exception, and for CDN$124.00 or U.S.$100.00 delivered, it remains an outstanding deal. A complete review, discussion of the DOS, DOS utility listings, and a comparison with the AERCO FD-ZX will appear on these pages in an upcoming issue. The system is available from Larken Electronics, RR#2 Navan, Ontario, Canada K4B 1H9.

EPROM SERVICES PROGRAMMER

I recently received a sample unit of the EPROM SERVICES PROGRAMMER MK.1, sent for evaluation by club member Larry Chavarie, of Ottawa, Ont.

After using this device, I decided I liked it enough to order one for myself. Although the unit is costly compared to the JLO programmer, the features justify the price.

As opposed to the JLO unit, which is memory mapped, the E/S programmer is completely port-mapped. The programmer, like the JLO, is a male card peripheral designed to plug into a motherboard. It measures approx. 3.6"w * 3.5"h, with a 28 pin Zero Insertion Force socket. An 8-pole DIP switch next to the ZIF socket for the EPROM permits selection of Intel type 2716, 2732, 2732A, 2764, 27128, and TI type 2516 & 2532 EPROMS.

The programmer is centred around an 8255 PIA chip, giving port-mapped control of all EPROM pins. An onboard switching boost-regulator provides the 21V or 25V programming voltage without the need for a separate power supply. Switching of the programming Vp is done automatically under software control. Eproms may be inserted or removed safely without powering down, permitting several EPROMS to be programmed without having to power down & reload between each one.

E/S PROGRAMMER, cont.

The truly remarkable part is the software. 1K (ZX81) or 1.5K (Spectrum/2068) in length, the MC software is available in 11K-12K or 31K-32K versions for the ZX81, and 30.5K-32K or 62K-63.5K versions for the Spectrum or Spectrumized 2068.

An 11-option Menu provides choices of: *TEST (compares each bit in EPROM with code to be programmed- it may be possible to overwrite a programmed EPROM), *COMPARE (an area of EPROM with an area of memory), *COPY (an area of EPROM into memory), *CLEAR (an area of memory to FF hex), *PROGRAM (see later), *CLEAN (check that EPROM is erased), *REPEAT (program another EPROM, same parameters), *E.DUMP (hex listing of EPROM), *M.DUMP (hex listing of memory), *PARAMETERS (lets you check before REPEATing), and QUIT (to BASIC).

When performing a PROGRAM, or any of the EPROM checking operations, parameters must be entered for start address of DATA, starting address in EPROM, and LENGTH of data. Entry is in hexadecimal format. When PROGRAMMING an EPROM, menu selection is made of EPROM type, parameters are entered, and the program takes over. First, the parameters are tested for validity. Then the EPROM is TESTed to see if it can be successfully programmed. If all is well, the programming Vp is switched on and programming commences. During programming, each byte is checked to see if programming is necessary, and if so, up to 4 attempts are made to program it. A verify is performed after each attempt. If the verification/retry fails, the program returns to the menu with a report of the failed address. Programming may be aborted at any time, with a report of the address about to be programmed.

I have found the EPROM SERVICES Programmer 1 to be very convenient and reliable. The capability of programming 27128 EPROMS with none of the complications inherent with a memory-mapped programmer is also a definite plus. The ZIF socket, switching flexibility, self-contained Vp source, and fancy software, as well as the ability to change EPROMs "on the run" makes this a very impressive unit. All this luxury doesn't come for free, though.

The Programmer 1 sells for£64.95 , from EPROM SERVICES. 3 WEDGEWOOD DRIVE, LEEDS LS8 1EF, ENGLAND.

MODified Shapes For T/S

This programming feature is adapted (well OK, cribbed) from the article "MODified Shapes for IBM", in COMPUTE!, May 1986 issue, written by Paul W. Carlson. This month, we'll discuss the background, and cover 1 of 4 programs which create nifty geometric patterns.

The program listing provided here will run on the ZX81/TS1000 with Oliger/TI Video and Silicon Mountain's JOBASIC or PIXL-ATR. It will also run on the 2068, but with reduced colour resolution.

The object of Mr. Carlson's article was to illustrate some interesting uses of the MOD (Modulo) command in IBM BASIC.

MODified Shapes, cont.

"MOD gives the integer (whole number) remainder of an integer division. For example, 17 MOD 3 = 2, because 17 divided by 3 equals 5, with a remainder of 2. Likewise, 30 MOD 5 = 0, since 5 divides evenly into 30 with a remainder of zero.

"Although some dialects of BASIC don't include a MOD operator, the INT function can be used for the same purpose. In Microsoft (or Sinclair) BASIC, the expression K-INT(K/J)*J gives exactly the same result as the IBM BASIC expression K MOD J.

"One of the most common uses of MOD is to test whether a value is odd or even. The expression X MOD 2 yields a 1 if X is odd, or 0 if X is even."

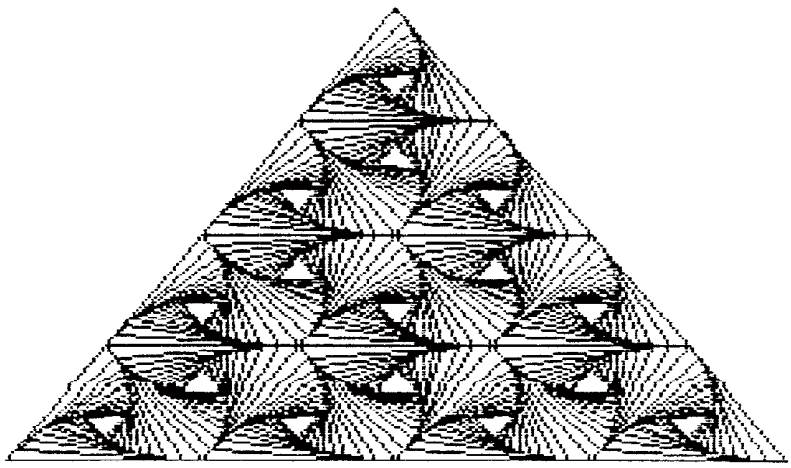
Next issue we'll discuss the use of Modulo arithmetic in further depth, and examine the operation of the following program.

```
  1 REM MODified Triangles for Timex-Sinclair
  2 REM For ZX81: INITIALIZE JOBASIC OR PIXL-ATR VARIABLES
  3 GOTO 40
  5 REM MODULO SUBROUTINE
  6 LET RES=ARG-INT (ARG/MOD)*MOD
  7 RETURN
 10 REM IBM> TI/2068 PLOT SCALING
 15 LET XX1=(X1/1.25)+XSET
 20 LET XX2=(X2/1.25)+XSET    •2068: USE
 25 LET YY1=191-(Y1/1.1)        YY1=175-(Y1/1.3)
 30 LET YY2=191-(Y2/1.1)        YY2=175-(Y1/1.3)
 35 RETURN
 39 REM TRIANGLE ROTATION
 40 DIM X(3)
 50 DIM Y(3)
 60 DIM Z(3)
 70 DIM T(3)
 80 LET XSET=1
100 LET SU=.1
110 LET RU=1-SU
120 LET II=1
130 LET C=1
200 FOR J=0 TO 3              • LOOP THRU J COLUMNS
210 LET II=-II                  AND I ROWS
220 LET JJ=1
230 FOR I=0 TO 6
240 LET JJ=-JJ
250 IF I<J OR I>6-J THEN GOTO 1100
300 IF J<2 OR I>2 THEN GOTO 320
310 GOTO 400
320 LET ARG=C                 • ADVANCE COLOUR MOD 3
330 LET MOD=3
340 GOSUB 5
350 LET C=RES+1
400 IF J=3 THEN GOTO 420
410 GOTO 500
420 LET ARG=C                 • ADVANCE COLOUR MOD 3
430 GOSUB 5
440 LET C=RES+1
```

*Can also be adapted for std. ZX81 or TS1000 with Callisto Software's GRAPHICA, or N. Elmaleh's SW HI-RES.

MODified Shapes, cont.

```
500 LET X(1)=0
510 LET X(2)=39
520 LET X(3)=78
530 LET Y(1)=0
540 LET Y(3)=0
550 IF II=JJ THEN GOTO 580
560 LET Y(2)=-48              • CHANGE Y(2) VALUES TO PLOT
570 GOTO 600                    SOME POLYGONS UPSIDE DOWN
580 LET Y(2)=48
600 FOR N=1 TO 11             • LOOP THRU N ROTATIONS
610 LET X1=3+X(3)+I*39
620 LET Y1=175-Y(3)-J*48+II*JJ*24
700 FOR M=1 TO 3              • LOOP THRU M SIDES
710 LET X2=3+X(M)+I*39
720 LET Y2=175-Y(M)-J*48+II*JJ*24
730 LET C=ARG
740 GOSUB 5
750 LET C=RES+1
800 GOSUB 10
802 REM SET INK COLOUR BY VARIABLE C
803 REM C SEQUENCES MOD3 +1, IE 1,2,3,1,2,3,1...
809 REM DRAW SIDE (JOBASIC):
810 IF USR V THEN LPRINT O;XX1,YY1;XX2,YY2
814 REM DRAW SIDE (2068):
815 PLOT XX1,YY1:DRAW XX2-XX1,YY2-YY1
820 LET X1=X2
830 LET Y1=Y2
840 LET ARG=M
850 GOSUB 5
860 LET NJ=RES+1
900 LET Z(M)=RU*X(M)+SU*X(NJ)
910 LET T(M)=RU*Y(M)+SU*Y(NJ)
920 NEXT M
1000 FOR P=1 TO 3             • MOVE RELATIVE COORDS
1010 LET X(P)=Z(P)              FROM "NEXT" ROTATION
1020 LET Y(P)=T(P)              TO "CURRENT" ROTATION
1030 NEXT P
1040 NEXT N
1100 NEXT I
1110 NEXT J
1200 IF INKEY$="" THEN GOTO 1200
1210 STOP
```



QUILLing ON THE QL

by Harry and Larry Sadler
R.R. #3 Kingston, Ont. K7L 4V2

I have had a QL for 10 months now and I thought other users and perhaps other Members considering a QL might be interested in my experience so far . This commentary will be restricted to QUILL as word processing is the main use I have made of my system . For the record, my system is configured with a JIL 12" green monitor, and a Smith-Corona L-1000 daisy wheel, friction feed, 12 cps printer . The printer isn't fast but it does for my prime purposes - correspondence .

I have been able to use four other word processors: WordStar, VolksWriter, Multimate, and Word Perfect . VolksWriter is simpler to use but less powerful . The other 3 are more powerful but harder to learn and to use . On a performance per dollar basis, QUILL wins easily .

QUILL has the feature that all the format intructions are filed with the document . This gives rise to the opportunity to creat a standard format to your liking that is different from the default format . For example, I like block format . The default format gives a paragaraph indent of 5 spaces, left margin at column 10, right at column 70, top maragin of 6, bottom margin of 3, and it is set to print page numbers . I prefer, margins at columns 5 and 75, no page numbers, and with a top margin of 1 line . I use single sheets so I can set my top margin at my convenience . Using a top margin of 1 line ensures that the print head will locate to the left maragin for the first line to be printed . These changes are made with the Design command . With the format that I like defined, I then save the blank document (with the new format set) in a document I call Setup . Thus to start a new document, I simply load my saved file called Setup and begin my text . If I am keeping the new document, I give it a new name when saving it to the microdrive .

Using block format (Left maragin and Indent margin at the same column) has created one interesting problem . If you go to a new paragraph and want to indent a few spaces, the cursor will not move . It will respond to characters other than the spacebar and you can tab into the new paragraph . My guess is this semi-lockup happens because QUILL expects a text character as the first character from the Indent Margin . This is logical in most cases but not when using block format . Another problem of getting the cursor to go where you want it is when I am using a right margin greater than 80 columns . Sometimes the cursor drops down a line instead of proceeding accross the former maximum screen width . I suspect something isn't quite right as it doesn't happen all the time . I get around it by using the left cursor key from the line below where I was working and had the trouble .

All in all, QUILL is very good . I have sent a few suggestions to Sinclair Research for the next version of QUILL. It will be intersting to see what happens next.

Here's looking forward to you trials & tribs and solutions!

BOB'S NOTEBOOK

Here is a short machine code program that very quickly converts all the numbers in a program to VAL expressions thereby saving quite a few bytes. I have found it very useful when trying to make room in a long program to add one of the LDOS routines used in the Larken Disk Drive system.
Type it in, run it and use it by the command RANDOMIZE USR 63100.

```
    1 REM COMPACTOR
    5 REM run and use RANDOMIZE
          USR 63100
   10 FOR i=63100 TO 63221: READ
a: POKE i,a: NEXT i
  100 DATA 42,83,92,43,237,75,75,
92,35,167,237,66,9,200,35,35,78,
35,70,229,35,126,254,13,32,3,209
,24,231,254
  110 DATA 14,32,243,209,213,11,1
1,11,197,120,18,27,121,18,229,43
,126,254,47,40,19,254,46,56,15,2
54,196,40,4,254
  120 DATA 58,48,7,35,35,119,43,4
3,24,231,35,54,176,35,54,34,225,
35,35,54,34,35,84,93,213,35,35,3
5,229,42
  130 DATA 89,92,167,237,82,68,77
,225,237,176,42,75,92,43,43,43,3
4,75,92,42,89,92,43,43,43,34,89,
92,225,193,24,155,0,0,0
  199 STOP
  200 SAVE "COMPACTOR": VERIFY ""
```

Speaking of LDOS, here are a few tips to the growing number of Larken users.

REMDOS may be used by saving it first on tape, then MERGEing it with another program (using the MERGE command on the TS2068). Be sure that Lines 1 and 2 are free for the REMDOS program.

Then call it using RANDOMIZE USR 26720. This is most useful when you are trying to save one of your programs to disk that has some code or data residing above 61000. If space is tight, it is here that COMPACTOR may come in handy, freeing up the necessary bytes.

Disk Menus.

When I build a menu for each of my disks, I try to add a routine in each of the programs on that disk that will automatically return to the menu for another selection; eg, if the program has a 'Quit' option that results in a STOP action, then replace that line with something like this:

```
CLS: OUT 64,64: RANDOMIZE USR
63488: REM load "menu.B1"
```

This is fairly easy to accomplish in BASIC but not so easy in machine language. So far I have been able to return only the LDOS keyboard input routine from within assembly language code. Larry Kenny talks about the DOS checking the value in address 23622 and if it is 255 then the action is as described above. Perhaps someone will know the solution to this one.

Cameron Hayne's Basic Compiler.

So far, it has performed very well. I have saved it to disk leaving out the logo which is frustrating on tape. I have put two games into machine code with excellent results. Others include a routine to relocate m/c to another start address and several calculating routines which move along very quickly. So it seems that Cameron has come up with a first rate utility program. Congrats to him!

Saving arrays to Disk.

Be sure to use one of the RAM versions of LDOS when you are trying to save and load arrays. With the Eprom DOS, arrays must be located first in the variables area and if they move to a different location when redimensioned during running, they cannot be located. The other LDOS's will find the arrays no matter where they are. Also, the other LDOS's allow the use of variable names to identify the text being saved. Saving an array t$ with a name n$ previously defined as 'tips' would require a line: LET n$="t$"+n$+".Ax" then a line, say: RANDOMIZE USR 26720:REM save n$.

Bob Mitchell June 1986.

TS 2068 TIP

To change the amount of time needed to hold down a key until it repeats, POKE 23561,X, where X may be a value between 1 and 35. Intially, on power-up the value will be 35. Each lower value will speed up by 1/60th of a second the interval untilthe key repeats

To change the amount of time between consecutive repeats, once the key starts repeating, POKE 23562,X. X will equal a value between 1 and 5. The lower the number the quicker the response will be.

TS 2068 SCROLLING TIP

The address 23692 is used by the computer to store a value used for SCROLL control. POKEing a number from 1 to 255 into this address will cause the computer to scroll that many lines + 21 on the screen.

USING THE TS 2068 WITH AN EXERCYCLE MACHINE

by Fred Schakel

The object of the whole experiment was to play a game on a computer (2068 or ZX81) and exercise at the same time.(I am in need of both!!) By combining an exercise bicycle, a joystick, and a 2068 with a special program, I proved that this is very possible.

My exercise bike already had a magnetic sensor on it, so I wired it in parallel with the FIRE button on my cursor joystick. The program displays a crude map of the subdivision that I live in, with a bicycle at my home, and several places of interest labelled as follows:

CTC --- Canadian Tire
MALL -- Shopping Mall
MOM --- Mother's place
A & D - Angie and Dwight's place
PK ---- Park
CLN --- Cleaners
MAC --- Mac's Milk store
H ----- My home
The blue area is the river

A machine code routine senses the bike speed through the FIRE button, by reading the bit on the zero key. This is much quicker than with BASIC. Even if the bike is stopped with the magnet on the switch, it recognises that the wheel is not turning. One may think that this should give a repeating "zero" key press, but this is so only in BASIC.

Getting away from the technicalities, the pink locations on the map represent goals to reach. When you reach one of them, points are awarded. It is also capable of letting you know when you go off the road or in the river. Steering is by left and right on the joystick, and you need to be moving in order to turn.

My plan is to involve obstacles such as pedestrians, trains, cars, puddles, etc. On the hardware end of it, I want to use an I/O port that will drive a motor which increases the tension on the bike. This would simulate going up and down hills.

I have included the program for you to try. (Just tap the zero key, and steer with keys 5 and 8 at the same time. It takes quite a few taps on the zero key to move ahead, depending on your speed. I will continue with the software and hardware development of this project, but anyone is welcome to help out. I can supply further information on the machine code for anyone who wants it.

An easy way to convert the speed signal from an exercycle to the 2068 is with a simple magnet and reed switch. Mount the magnet near the edge of the wheel. Put the reed switch on the cycle frame so that, as the wheel turns, the magnet passes close by the switch without touching it. Wire the reed switch leads to the two contacts on the fire button of the joystick. The joystick port can now read the speed of the wheel similarly to a tachometer.

Listing for X-R-Cycle Program

```
  10 LET o=0
  20 POKE 58443,74: POKE 58444,2
01: POKE 58445,0: POKE 58446,64:
 RANDOMIZE USR 51480: REM move
screen from 51530 to d-file
  50 REM  init vars
  60 LET a=2: LET x=12: LET y=12
  70 LET sc=0: LET x1=x: LET y1=
y
  80 GO SUB 9200: LET a$="↑→↓←"
  90 PRINT AT 12,11; INK 3; PAPE
R 0;"■";AT 12,4;" ";AT 5,19;" ";
AT 12,28;" "
 100 REM  print bike
 110 PRINT AT y,x; INK 6; PAPER
0;"■"
 130 PRINT INK 4; PAPER 6;AT y1,
x1;a$(a)
 140 LET x=x1: LET y=y1
 200 REM sense speed
 210 LET w=USR 51400: IF w<20 OR
 w>1000 THEN LET s=0: GO TO 260
 220 LET s=1500/w
 260 PRINT AT 20,1;"SPEED ";INT
s;" "
 300 REM direction
 310 LET a=a+(IN 61438=28)-(IN 6
3486=16): IF a=0 THEN LET a=4
 320 IF a=5 THEN LET a=1
 400 REM locate
 405 LET d=s/500
 410 LET x1=x1+(a=2)*d-(a=4)*d
 420 LET y1=y1+(a=3)*d-(a=1)*d
 430 LET o=o+d
 440 PRINT AT 21,1;"MILES ";INT
o;" "
 500 REM collision
 505 LET c=ATTR (y1,x1)
 510 IF c=6 OR c=62 THEN GO TO 1
00
 520 IF c=5 THEN PRINT FLASH 1;A
T 20,12;"FELL IN RIVER": GO TO 7
00
 530 IF c=3 THEN GO SUB 1000: LE
T sc=sc+p: PRINT AT 20,12;"ARRIV
ED at ";q$;AT 21,19;"SCORE ";sc:
 GO SUB 2000: GO TO 130
 540 PRINT AT 20,12;"RAN OFF the
 ROAD": GO TO 700
 700 FOR q=1 TO 4: BEEP .1,11: B
EEP .1,33: NEXT q: PAUSE 150: GO
 TO 80
1000 IF y1>17 THEN GO TO 1100
1010 IF INT y1=12 THEN GO TO 120
0
1020 IF INT y1<7 THEN LET q$="PA
RK": LET p=425: IF INT y=5 THEN
LET a=3: RETURN
1025 LET a=2: RETURN
1100 IF INT x1=4 THEN LET q$="MO
M'S": LET p=350: LET a=1: RETURN

1110 IF INT x1=10 THEN LET q$="M
AC'S": LET p=225: LET a=1: RETUR
N
1120 IF INT x1=21 THEN LET q$="C
LEANERS": LET p=175: LET a=1: RE
TURN
1130 IF INT x1=28 THEN LET q$="C
TC": LET p=550: LET a=1: RETURN
1200 IF INT x1=4 THEN LET q$="MA
LL": LET p=350: LET a=3: RETURN
1210 IF INT x1=11 THEN LET q$="H
OME": LET p=0: RETURN
1220 IF INT x1=27 THEN LET q$="A
&D": LET p=550: LET a=3: RETURN
2000 LET y1=y: LET x1=x: FOR q=1
 TO 50: LET w=USR q: NEXT q: PRI
NT PAPER 0;AT 20,12;"
           ": RETURN
```

cont'd on next page

```
9000 CLEAR 51000: LOAD ""CODE :
BEEP .2,22: RUN
9199 REM get map from memory
9200 DATA 16,124,84,56,56,56,16,
16,0,0,8,58,255,58,8,0,16,16,56,
56,56,84,124,16,0,0,96,92,255,92
,96,0
9210 RESTORE 9200: FOR q=65496 T
O 65527: READ w: POKE q,w: NEXT
q: RETURN
9999 SAVE "bike" LINE 9000: BEEP
.01,11: SAVE "code"CODE 51400,7
045: BEEP .01,11
```

```
NOTE: About 7K of code is also
required to run this program
```

Fred wrote to me about this project, and I thought it was so interesting it should be put into the newsletter. I am sure Fred would like to hear from anyone out there so I take the liberty of giving his address. Fred Schakel, 181 Seawood Ave. London Ont. N6J 1B8

George Chambers

LARKEN USER NOTES

These helpful tips on LARKEN DD operation have been supplied to us by Larry Kenny.

- If you have CRC errors usually on the inside tracks of your disk, the problem may be that the variable resistor on the controller board needs to be adjusted slightly. This adjustment is to compenate for variations on the pulse widths of the "read data" on different disk drives. (This trimmer resistor is near the 74LS00 on the controller board).

Remember where the adjustment is initially set because it should only have to be moved a small amount. If it is a single-turn trimmer then you should only have to turn it less than 1/4 turn in either direction. If it is a 10-turn trimmer (metal screw) then you should only have to turn it 1 or 2 turns.

- The CLEAR 61000 command before entering the LDOS is usually not necessary if you are in the TS2068 mode. The Spectrum has to have RAMTOP changed by a CLEAR command because the Z80 stack is there and it has to be moved to make room for the LDOS. It is a good idea, when saving programs on the Spectrum, to always have RAMTOP set in a standard place (e.g. 61000). This way you won't have to remember a lot of different CLEAR commands when loading. After the program loads it can set RAMTOP to where it needs it.

- If your program uses UDG's (user defined graphics), you must make sure that the EPROM is not in the top of memory (shadowing the UDG's memory area) by doing an OUT 84,0 command.

- If you have a very large BASIC program (extends past 61000) that you would like to put onto disk, an easy way is to save a copy of LDOS16 to tape. Then load in your large BASIC program. If it is an auto-run BASIC program, to stop it, MERGE it instead of loading it. Then add these lines to the program in an appropriate place.

```
9000 LOAD "LDOS16": RANDOMIZE USR
    16384: REM save "Bigprg.B1"
9010 GOTO (start)  (or RUN)
```

Then type GOTO 9000 (or wherever you put it) and it will load LDOS16 off the tape, and then save the program to disk.

- To load the big program, you can't directly use the EPROM DOS because the top of the program will overwrite the LDOS, so you need a BASIC loader. e.g.:

```
10 OUT 84,64: RANDOMIZE USR
   63488: REM save "Basloa.B1"
20 RANDOMIZE USR 63488:
   REM load "LDOS16.C5"
30 RANDOMIZE USR 16384:
   REM load "Bigprg.B1"
```

This program when run will save itself in auto-run mode.

- Note that with Sinclair BASIC you cannot put an active command in a line after a REM statement. e.g.:

```
RANDOMIZE USR 63488: REM save "xxxxxx.B1": RUN
```

The RUN following the REM statement will be ignored.

- If you would like to print your directory on the printer, you can use the LPRINT USR command instead of PRINT USR when entering the LDOS. You won't be able to see what you are typing until you press ENTER.

- Sometimes within running programs, you may get a "NO FILE" error; maybe because you changed your program or disk name. The best way to return to your original program is to keep pressing ENTER until the "scroll ?" prompt, then BREAK it. This method is preferable because if you were going to load a code program and then jump to it after it was loaded; if the LDOS came up with "NO FILE" and you typed "EXIT", the program would crash.

- You may have a machine code utility or routine such as a printer driver that you use with your BASIC program. If you place it anywhere between address 23552 (start of system variables) and the address of VARS, it will be saved with your program. This also applies to any changes to the streams and channels tables.

Retyped by George Chambers

ZX81 SCREEN ADDRESSING

by Mike Lemyre

The print position address is found at PEEK 16389 + 256 * PEEK 16399.

If the program area is empty then this address = 16510.

If this address is POKEd with a character code then the referenced character will be displayed at position 0,0.

To print a line, POKE this address with the code in increments of 1 to a maximum of 31.

The 33rd print position must hold code 118 page number.

PEEK 16441 for the X value and PEEK 16442 for the Y value to find the current print position.

To find the PLOT co-ordinates:

PEEK 16438 for X

PEEK 16439 for Y

To calculate the address of the screen the starting addres can be called X. For the sake of argument we will call the starting point 0,0.

Each additional point will equal X + 1. The maximum value to be added = 31. Each additional line is 33 bytes from the previous line. An example is:

Let X = 16510, therefore line 1 would end at 16541 and line 2 would start at 16543. 16510 is the equivalent address as the print at 0,0.

If a loop is used to fill the screen with digits then the loop must have a counter to maintain the correct parameters of the print position.

THE SCREEN ADDRESS VALUES

0,0	0,31
1,33	1,64
2,66	2,97
3,99	3,130
4,132	4,163
5,165	5,196
6,198	6,229
7,231	7,262
8,264	8,295
9,297	9,328
10,330	10,361
11,363	11,394
12,396	12,427
13,429	13,460
14,462	14,493
15,495	15,526
16,528	16,559
17,561	17,592
18,594	18,625
19,627	19,658

TS2068 SCREEN ADDRESSING

by Mike Lemyre

To print a digit on the 2068 you must PEEK 23684 + 256 * PEEK 23685. This gives the starting addresss of the display file and must be increased by increments of 256 up until a value of 2048. To print more than one digit you can increase the original address after storing it in a register, thus it can be referenced later to allow the print position to be altered.

TO PRINT A DIGIT ON THE 2068

```
10 LET n=PEEK 23684 + 256 * PEEK 23685
20 POKE n,255
30 LET n = n + 256
35 IF n>= 18432 THEN STOP
40 GO TO 20
```

PEEK 23688 gives the X value of the print position.

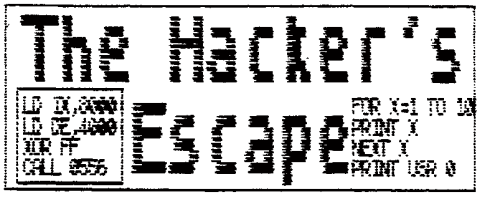
PEEK 23689 gives the Y value of the pront position.

Note that the print position 0,0 will have the value of 33,24. Position 20,0 will have the value of 33,3.

The screen address coordinates start at 0,0 in the lower right hand, and end at 33,24 in the upper left.

Both the ZX81 and the 2068 have the same display file layout.

```
10 LET n = PEEK 23684 + 256 * PEEK 23685
12 LET k = n
14 LET c = 0
20 FOR m = 1 TO 8
30 POKE n,255
40 LET n = n + 256
50 NEXT m
52 LET c = c + 1
55 LET n = k + c
67 IF c>=20 THEN GO TO 95
70 GO TO 20
95 LET c = 0
100 LET k = k + 32
105 LET n = k
109 IF c>= 20 THEN STOP
110 GO TO 20
```



Welcome to the Hackers Escape. A new column featuring ways of going about making back-up copies of your precious software, and questions related to making back-up copies of programs. This column is FOR YOUR OWN PERSONAL USE and not for pirating programs or distributing them.

There are four ways of going about making back-ups of programs, and each has it's own advantages and disadvantages. The first method of backing up a program is tape-to-tape. Tape-to-tape is good for making back-ups of those long programs that occupy the whole memory. But on the other hand it is practically useless for making back-ups of fastloaded programs. This is usually due to incorrect head alignment and/or improper cleaning of playback head, erase head, pinch roller and capstan. It is advisable to use two recorders of the same make and model when making tape-to-tape back-ups. Sometimes the copied program will not even register on the screen, or shows an R tape loading error after the loader program.

Dubbing cords or attenuation cords are good, but care has to be taken if the cord over- or under-attenuates. What the attenuating cord does is take the signal coming from the EAR jack of one of the tape recorders and reduce it to a preset level (depending on the capacitance of cord). This filters out any excessive background noise. The signal is then filtered, giving a sharper tone (again depending on the cord make-up). The other tape recorder then records the signal coming in the MIC jack. Over-attenuationg cords tend to cut crucial data signals that result in program errors. Under-attenuating cords dull the sound and allow more noise to creep in.

The second method of backing up a program is with a tape copier. Tape copiers are very popular for making back-ups of software. Care should be taken when using a tape copier. The advantage of the copy program is that it enables you to copy another program by loading it into the computer, and simultaneously sending it back to a second recorder with a refreshed signal. Copiers are very error-prone and should be used with great care, because the copy made by the program may not work, or even load correctly. Always use a tape copier with the latest features that cope with the software you are using. Always be aware of the single long 48K file types that cannot be backed up manually. Make sure your program can cope with false headers and headerless files.

The third method of making back-ups. This method is extremely unique and very easy to use. This type is not software, but hardware. The interface TOTALLY bypasses all types of copy-protection in the program, no matter how heavily it is protected. What it does is make a complete dump of the memory to tape, without headers. After the SAVE, you simply load in the software supplied with the interface, and convert the headerless files to tape or to microdrive.

Finally the fourth method. You may call it a method if you are the type that is ambitious and are completely familiar with the computer. This method involves manually breaking into the program, and decoding the file of hidden machine code or hidden program lines. This can be very tedious and stressful if you are dissecting the program, but always keep in mind that there are the other three ways of saving programs.

If you have any questions relating to this or just need help, write to Karim Rahamet, 101 White Oaks Court, Pent-house #7, Whitby, Ontario, L1P 1A1.

All questions will be answered by mail, or by phone if given.

MURPHY'S LAWS OF COMPUTING

People always remember the last mistake you made

He who hesitates is probably smart.

The one who does the least work always gets the most credit.

The less a peripheral system costs, the more it costs to fix.

Whatever happens, behave like you meant it to happen.

Confidence is the feeling you get just before you fully understand the problem.

When you get to the point where you really understand your system, it's probably obsolete.

As soon as you find that your computer is easy to use, add some peripherals you don't understand how to operate.

No matter what goes wrong with your carefully planned database system, there's always someone who says they knew it would.

It's only when you need to knock on wood that you realize that the world it entirely made up of aluminum and plastic.

DISK DROPPINGS
by Greg Lloyd
Compuserve ID 76317,1461

In the previous column, I looked at the bits of hardware needed for the disk interface. This time we'll look at the whys and hows.

First to answer the Whys. Why not! Are you fed up with the slow error-prone cassette system Sir Clive bequeathed to his serfs? It doesn't matter how careful you are, everyone has lost a program or a file due to a tape glitch. I'm not saying you won't lose any programs on disk, but you will know in a few seconds if you have a problem, rather than a few minutes on tape.

Since most of the software written for the 2068 and Spectrum uses memory to the fullest, your Disk Operating System should use little or no RAM. This is the theory, but in practice there are few systems that live up to this principle. You have to use your noodle to develop ways of fitting tape-based software into your disk system. I found that a header reader program is a must for any memory-mapping. If you know where all the pieces should be, you can fit them all together when you save them on disk. A real plus of a disk system is that you no longer have to be concerned with RAM size. You can write programs that use the disk as memory. IBM call this virtual memory because if you have enough disks it can go on forever.

Here's an example of how it is done. Imagine a filing program that stores all the uses of a dead EPROM chip in r$(1000,20). This could mean that you are limited to 1000 records, holding 20 characters of data. When that is used, POOF! ERROR 4 OUT of memory. But with our clever program that draws it's information from the disk, you can have as many records as you wish. The other uses of this technique are for downloading those big files from Compuserve (Groliers Encyclopaedia), or using your Word Processor to write the Great White Northern Novel. It is possible to wear out your disk drive and your arm long before you reach the limits of this technique.

My system, from Larken Electronics, has the ability to load and save an array of data or a block of code. This simple fact can be used to implement the virtual memory idea. Other systems allow the use of sequential and random access files. This is one step beyond saving and loading an array or code. It allows the records to be stored and retrieved one after another sequentially (sometimes slowly), or in a partitioned block for easy access randomly (fast). For a small file, sequential is probably best. Larger files dictate the use of random access.

The only limits you have with your Disk System are the ones you place there. If you let your imagination go, your system can do anything you want it to do. Ask someone like Peter McMullin (our resident ZX81 Forever and Ever and Ever Person) if you doubt me. 08/03/86

This is a page from the British ZX Spectrum manual, which is a bit more informative than the TS2068 manual. Did you realize that the top row of the 2068 keyboard provides 77 functions!!!

Fred Schakel, London T/S Users Group.

MODE	SHIFT										
E	SYMBOL	DEF FN	FN	LINE	OPEN #	CLOSE #	MOVE	ERASE	POINT	CAT	FORMAT
	CAPS	Ink blue	Ink red	Ink magenta	Ink green	Ink cyan	Ink yellow	Ink white	Flash off	Flash on	Ink black
	NONE	Paper blue	Paper red	Paper magenta	Paper green	Paper cyan	Paper yellow	Paper white	Normal brightness	Extra bright	Paper black
G	EITHER	▛	▜	▄	▙	▌	▚	▗	■	EXIT GRAPHICS	DELETE
	NONE	▘	▝	▀	▖	▐	▞	▟	□	EXIT GRAPHICS	DELETE
K,L or C	CAPS	EDIT	CAPS LOCK	TRUE VIDEO	INVERSE VIDEO	←	↓	↑	→	GRAPHICS MODE	DELETE
	SYMBOL	!	@	#	$	%	&	'	(	)	_
	NONE	1	2	3	4	5	6	7	8	9	Ø

NEW PRODUCTS ANNOUNCEMENT
For TS2068 and SPECTRUM

NovelSoft *PROUDLY PRESENTS THE ALL NEW...*

Artworx

VERSION 1.1

* Pull-Down menus
* Several Brushes
* Spray
* Auto-Fill
* Zoom
* Undo
* Several Text Fonts
* Cut & Paste Windows
* Auto-Speed Control
* Magnify & Reduce
* Rotate & Mirror
* Full Attribute Control
* Fully Elastic Shapes including Circle, Box Triangle, Ray and Line
* Fast Ellipse and Arc
* Includes Spectrum & TS2068 Versions
* Supports Microdrives and Kempston Joystick
* Includes GALLERY, the slide show/animator
* 5 Samples of Artwork
* Excellent Manual

Unshackle your creativity with... *ARTWORX!*

ARTWORX V1.1-$19.95 U.S. plus $3.00 S&H

NovelSoft *INTRODUCES THE REVOLUTIONARY NEW BASIC COMPILER...*

TIMACHINE

The dream of every BASIC programmer has now been realized!

* TIMACHINE will turn your BASIC into super-fast machine code, running up to 200 times faster!
* Runs faster than PASCAL
* Handles floating point operations like SIN, COS, TAN
* Handles all BASIC except I/O
* Includes an excellent manual and 4 demonstration programs
* Compiles up to 27K in seconds
* Includes Spectrum & 2068 versions

Super-Charge your BASIC programs with...*TIMACHINE!*

TIMACHINE -$19.95 U.S. plus $3.00 S&H

NovelSoft
A FORMAT FOR THE FUTURE

106 Seventh Street, Toronto, Ontario, Canada M8V 3B4 • Tel. (416) 259-8682 • CompuServe 70416,1435

LENSLOK

This representation of Lenslok shows how the 13 vertical stripes in the clear plastic lens are angled to act as prisms. Each prism sees a particular vertical line of pixels on the screen. Six of these lines create the image of one letter or number while another six create the second character. This is the diagram of the Lenslok that comes with "Tomahawk". Lensloks for other programs would not necessarilly have the same characteristics.

Fred Schakel
Timex/Sinclair Club
London, Ont.

Canada 1

Postmaster, if Undelivered Return to :

Toronto Timex - Sinclair Users Club
P. O. Box 7274 Stn. A
Toronto, Ont., M5W 1X9
Canada

To

[redacted]

6 09

3